Attention Allocation for Decision Making Queues[☆]

Vaibhav Srivastava[a], Ruggero Carli[b], Cédric Langbort[c], and Francesco Bullo[a]

[a]*Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, USA,* {vaibhav,bullo}@engineering.ucsb.edu
[b]*Department of Information Engineering, University of Padova , Italy,* carlirug@dei.unipd.it
[c]*Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA,* langbort@illinois.edu

Abstract

We consider the optimal servicing of a queue with sigmoid server performance. The sigmoid server performance occurs in various domains including human decision making, visual perception, human-machine communication and advertising response. The tasks arrive at a given rate to the server. Each task has a deadline that is incorporated as a latency penalty. We investigate the trade-off between the reward obtained by processing the current task and the penalty incurred due to the tasks waiting in the queue. We study this optimization problem in a Markov decision process (MDP) framework and show that the MDP formulation is equivalent to a certainty-equivalent problem. We determine the receding horizon servicing policy for the queue and show that the optimal policy may drop some tasks, that is, may not process a task at all. We then develop an adaptive policy that incorporates all the available information about the current tasks and show that the adaptive policy improves the performance significantly. Finally, we present a comparative study of the receding horizon policy for the certainty-equivalent problem and the adaptive policy. We also suggest guidelines for the design of such queues.

Keywords: optimal control of queues, non-submodular optimization, sigmoid utility, human decision making

1. Introduction

The recent national robotic initiative [10] underlines innovative robotics research and applications emphasizing the realization of co-robots acting in direct support of and in a symbiotic relationship with human partners. Such co-robots will facilitate better interaction between the human partner and the automaton. In complex and information rich environments, one of the key roles for these co-robots is to help the human partner efficiently focus her attention. A particular example of such a setting is the surveillance mission, where the human operator monitors the evidence collected by the autonomous agents [5, 7]. The excessive amount of information available in such systems often results in poor decisions by the human operator [23]. This emphasizes the need for the development of a support system that helps the human operator optimally focus her attention.

Recently, there has been significant interest in understanding the physics of human decision making [4]. Several mathematical models for human decision making have been proposed [4, 15, 27]. These models suggest that the correctness of the decision of a human operator in a binary decision making scenario evolves as a sigmoid function of the time-duration allocated for the decision. Thus, the probability of the correct decision by a human operator increases slowly for small time-duration allocations and high time-duration allocations, and increases quickly for moderate time-duration allocations. The sigmoid function also models the quality of human-machine communication [27], the human performance in multiple target search [12], the advertising response function [26], and the expected profit in simultaneous bidding [17]. Therefore, the analysis presented in this paper can also be used to determine optimal human-machine communication policies, optimal search strategies, the optimal advertisement duration allocation, and optimal bidding strategies. In this paper, we generically refer to the server with sigmoid performance as a human operator and the tasks as the decision making tasks. When a human operator has to serve a queue of decision making tasks *in real time*, the tasks (e.g., feeds from camera) waiting in the queue lose value continuously. This trade-off between the correctness of the decision and the loss in the value of the pending tasks is of critical importance for the performance of the human operator. In this paper, we address this trade-off, and determine the optimal duration allocation policies for the human operator serving a decision making queue.

There has been significant interest in the study of the performance of a human operator serving a queue. In an early work, Schmidt [21] models the human as a server and numerically studies a queueing model to determine the performance of a human air traffic controller. Recently, Savla et al [20] study human supervisory control for unmanned aerial vehicle operations: they model the system by a simple queuing network with two components in series, the first of which is a spatial queue with vehicles as servers and the second is a conventional queue with human operators as servers. They design joint motion coordination and operator scheduling policies that minimize the expected time needed to classify a target after its ap-

[☆]This work has been supported in part by AFOSR MURI Award FA9550-07-1-0528. A preliminary version of this work [25] entitled "Task release control for decision making queues" was presented at American Control Conference, 2011, San Francisco, CA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
22 FEB 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Attention Allocation for Decision Making Queues	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Barbara,Center for Control, Dynamical Systems and Computation,Santa Barbara,CA,93106**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We consider the optimal servicing of a queue with sigmoid server performance. The sigmoid server performance occurs in various domains including human decision making, visual perception, human-machine communication and advertising response. The tasks arrive at a given rate to the server. Each task has a deadline that is incorporated as a latency penalty. We investigate the trade-off between the reward obtained by processing the current task and the penalty incurred due to the tasks waiting in the queue. We study this optimization problem in a Markov decision process (MDP) framework and show that the MDP formulation is equivalent to a certainty-equivalent problem. We determine the receding horizon servicing policy for the queue and show that the optimal policy may drop some tasks, that is, may not process a task at all. We then develop an adaptive policy that incorporates all the available information about the current tasks and show that the adaptive policy improves the performance significantly. Finally, we present a comparative study of the receding horizon policy for the certainty-equivalent problem and the adaptive policy. We also suggest guidelines for the design of such queues.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

pearance. The performance of the human operator based on her utilization history has been incorporated to design maximally stabilizing task release policies for a human-in-the-loop queue in [19, 18]. Bertuccelli et al [3] study the human supervisory control as a queue with re-look tasks. They study the policies in which the operator can put the tasks in an orbiting queue for a re-look later. An optimal scheduling problem in the human supervisory control in studied in [2]. The authors determine a sequence in which the tasks should be serviced so that the accumulated reward is maximized. Powel et al [16] model mixed team of humans and robots as a multi-server queue and incorporate a human fatigue model to determine the performance of the team. They present a comparative study of the fixed and rolling work-shifts of the operators.

The optimal control of queueing systems [22] is a classical problem in queueing theory. Stidham et al [13] study the optimal service policies for a M/G/1 queue. They formulate a semi-Markov decision process, and describe the qualitative features of the solution. Certain technical assumptions in [13] are relaxed by George et al [8]. In contrast to the models discussed here, these studies assume identical tasks and submodular performance functions. Hernández-Lerma et al [11] determine optimal servicing policies for the identical tasks and some arrival rate. They adapt the optimal policy as the arrival rate is learned.

In this paper, we study the problem of optimal time-duration allocation in a queue of binary decision making tasks with a human operator. We refer to such queues as *decision making queues*. We assume that tasks come with processing deadlines and incorporate these deadlines as a soft constraint, namely, latency penalty. We consider two particular problems. First, we consider a static queue with latency penalty. Here, the human operator has to serve a given number of tasks. The operator incurs a penalty due to the delay in processing of each task. This penalty can be thought of as the loss in value of the task over time. Second, we consider a dynamic queue of the decision making tasks. The tasks arrive at a fixed rate and the operator incurs a penalty for the delay in processing each task. In both the problems, there is a trade-off between the reward obtained by processing a task, and the penalty incurred due to the resulting delay in processing other tasks. We address this particular trade-off.

The major contributions of this work are as follows: (i) we determine the optimal duration allocation policy for the static decision making queue with latency penalty; (ii) we pose an MDP to determine the optimal allocations for the dynamic decision making queue and show that the MDP formulation is equivalent to a certainty-equivalent problem; (iii) we provide a simple procedure to determine a receding horizon policy for the certainty-equivalent problem, namely, certainty-equivalent policy; (iv) we establish performance bounds for the certainty-equivalent policy; (v) we study an adaptive algorithm that incorporates all the available information about the current tasks and improves the performance of the certainty-equivalent policy; (vi) we present a comparative study of the certainty-equivalent policy and the adaptive policy; (vii) we suggest some guidelines for the design of decision making queues.

The remainder of the paper is organized as follows. We discuss some preliminary concepts in Section 2. We present the problem setup in Section 3. The static queue with latency penalty is considered in Section 4. We pose the optimization problems associated with the dynamic queue with latency penalty and study their properties in Section 5. We present and analyze receding horizon algorithms for these optimization problems in Section 6. A real time adaptive algorithm is studied in Section 7. Our conclusions are presented in Section 8.

2. Preliminaries

In this section, we present some concepts that are used throughout the paper. We start with some models of human decision making, followed by some properties of sigmoid functions. We close the section with a discussion on receding horizon optimization.

2.1. Speed-accuracy trade-off in human decision making

Consider the scenario where, based on the collected evidence, the human has to decide on one of the two alternatives H_0 and H_1. The evolution of the probability of correct decision has been studied in cognitive psychology literature [15, 4].

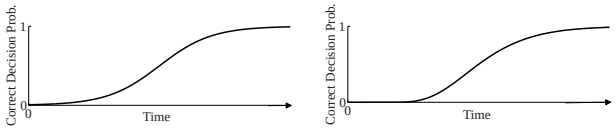
Pew's model: The probability of deciding on hypothesis H_1, given that hypothesis H_1 is true, at a given time $t \in \mathbb{R}_{\geq 0}$ is given by

$$\mathbb{P}(\text{say } H_1 | H_1, t) = \frac{p_0}{1 + e^{-(at-b)}},$$

where $p_0 \in [0, 1]$, $a, b \in \mathbb{R}$ are some parameters specific to the human operator [15].

Drift diffusion model: Conditioned on the hypothesis H_1, the evolution of the evidence for decision making is modeled as a drift-diffusion process [4], that is, for a given a drift rate $\beta \in \mathbb{R}_{>0}$, and a diffusion rate $\sigma \in \mathbb{R}_{>0}$, the evidence Λ at time t is normally distributed with mean βt and variance $\sigma^2 t$. The decision is made in favor of H_1 if the evidence is greater than a decision threshold $\eta \in \mathbb{R}_{>0}$. Therefore, the conditional probability of the correct decision at time t is

$$\mathbb{P}(\text{say } H_1 | H_1, t) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \int_{\eta}^{+\infty} e^{\frac{-(\Lambda-\beta t)^2}{2\sigma^2 t}} d\Lambda.$$



(a) Pew's model (b) Drift diffusion model

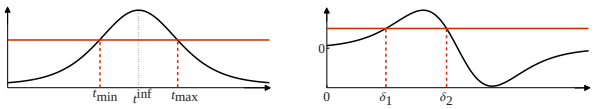
Figure 1: The evolution of the probability of the correct decision under Pew's and drift diffusion model. Both curves look similar and are sigmoid.

2.2. Sigmoid functions

A doubly differentiable function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$f(t) = f_{\text{cvx}}(t)\mathcal{I}(t < t^{\text{inf}}) + f_{\text{cnv}}(t)\mathcal{I}(t \geq t^{\text{inf}}),$$

is called a sigmoid function, where f_{cvx} and f_{cnv} are monotonically increasing convex and concave functions, respectively, $\mathcal{I}(\cdot)$ is the indicator function and $t^{\text{inf}} \in \mathbb{R}_{>0}$ is the inflection point. The derivative of a sigmoid function is a unimodal function that achieves its maximum at t^{inf}. Further, $f'(0) \geq 0$ and $\lim_{t\to+\infty} f'(t) = 0$. Also, $\lim_{t\to+\infty} f''(t) = 0$. A typical graph of the first and second derivative of a sigmoid function is shown in Figure 2. From the derivative of the sigmoid function, it is clear that the sigmoid functions are not submodular. Note that the evolution of the conditional probability of the correct decision is a sigmoid function in Pew's as well as drift-diffusion model.



(a) (b)

Figure 2: (a) First derivative of the sigmoid function and the penalty rate. A particular value of the derivative may be attained at two different times. The total benefit, that is, the sigmoid reward minus the latency penalty, decreases up to $t_{\min}$, increases from $t_{\min}$ to $t_{\max}$, and then decreases again. (b) Second derivative of the sigmoid function. A particular positive value of the second derivative may be attained at two different times.

2.3. Receding horizon optimization

Consider the following infinite horizon dynamic optimization problem:

$$\begin{aligned} &\text{maximize} && \sum_{\ell=1}^{+\infty} \psi(x(\ell), u(\ell)) \\ &\text{subject to} && x(\ell+1) = \phi(x(\ell), u(\ell)),\ x(0) \text{ given}, \end{aligned} \tag{1}$$

where $x(\ell), u(\ell) \in \mathbb{R}$ are the state and the control input at time $\ell \in \mathbb{N}$, respectively, $\psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the stage cost, and $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defines the nonlinear evolution of the system.

In receding horizon optimization [6], the optimization problem (1) is approximated by the following finite horizon optimization problem at each stage $\theta \in \mathbb{N}$:

$$\begin{aligned} &\text{maximize} && \sum_{\ell=\theta}^{\theta+N-1} \psi(x(\ell), u(\ell)) \\ &\text{subject to} && x(\ell+1) = \phi(x(\ell), u(\ell)),\ x(\theta) \text{ given}, \end{aligned} \tag{2}$$

where $N \in \mathbb{N}$ is a finite horizon length. The receding horizon optimization is summarized in Algorithm 1.

Algorithm 1 Receding horizon optimization

1: at stage $\theta \in \mathbb{N}$, observe state $x(\theta)$
2: Solve optimal control problem (2) and compute the optimal control inputs $u^*(\theta), \ldots, u^*(\theta + N - 1)$
3: Apply $u^*(\theta)$, and set $\theta = \theta + 1$
4: Go to step 1:

3. Problem setup

We consider the problem of optimal time duration allocation for a human operator. The decision making tasks arrive at a given rate and are stacked in a queue. A human operator processes these tasks on the *first-come first-serve* basis (see Figure 3.) The human operator receives a unit reward for the correct decision, while there is no penalty for a wrong decision. We assume that the tasks can be classified according to their difficulty, and the difficulty level takes value in an arbitrary set $\mathcal{D} \subseteq \mathbb{R}^q$, for some $q \in \mathbb{N}$. For a decision made after processing a task with difficulty $d \in \mathcal{D}$ for time t, the expected reward is

$$\mathbb{E}[\mathbf{1}_{\text{say } H_1 | H_1, t}] = \mathbb{P}(\text{say } H_1 | H_1, t) = f_d(t), \tag{3}$$

where $f_d : \mathbb{R}_{\geq 0} \to]0, 1[$ is the sigmoid function associated with the task. Note that such reward structure corresponds to the expected number of correct decisions.

We consider two particular problems. First, in Section 4, we consider a static queue with latency penalty, that is, the scenario where the human operator has to perform $N \in \mathbb{N}$ decision making tasks, but each task loses value at a constant rate per unit delay in its processing. Second, in Sections 5, 6, 7, we consider a dynamic queue of decision making tasks where each task loses value at a constant rate per unit delay in its processing. The loss in the value of a task may occur due to the processing deadline on the task. In other words, the latency penalty is a soft constraint that captures the processing deadline on the task. For such a decision making queue, we are interested in the optimal time-duration allocation to each task. Alternatively, we are interested in the task release rate that will result in the desired accuracy for each task. We intend to design a decision support system that tells the human operator the optimal time-duration allocation to each task.

Remark 1 (Soft constraints versus hard constraints)**.** The processing deadlines on the tasks can be incorporated as hard constraints as well, but the resulting optimization problem is combinatorially hard. For instance, if the performance of the human operator is modeled by a step function with the jump at the inflection point and the deadlines are incorporated as hard constraints, then the resulting optimization problem is equivalent to the N-dimensional knapsack problem [14]. The N-dimensional knapsack problem is NP-hard and admits no fully polynomial time approximation algorithm for $N \geq 2$. The standard [14] approximation algorithm for this problem has factor of optimality $N + 1$ and hence, for large N, may yield results very far from the optimal. The close connections between the knapsack problems with step functions and sigmoid functions (see [24]) suggest that efficient approximation algorithms may not exist for

the problem formulation where processing deadlines are modeled as hard constraints. □

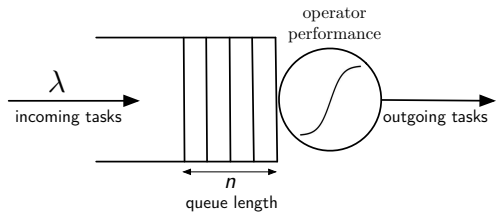


Figure 3: Problem setup. The decision making tasks arrive at a rate λ. These tasks are served by a human operator with sigmoid performance. Each task loses value while waiting in the queue.

4. Static queue with latency penalty

4.1. Problem description

Consider that the human operator has to perform $N \in \mathbb{N}$ decision making tasks in a prescribed order (task labeled "1" should be processed first, etc.) Let the human operator allocate duration t_ℓ to the task $\ell \in \{1, \dots, N\}$. Let the difficulty of the task ℓ be $d_\ell \in \mathcal{D}$. According to the importance of the task, a weight $w_\ell \in \mathbb{R}_{\geq 0}$ is assigned to the task ℓ. The operator receives an expected reward $w_\ell f_{d_\ell}(t_\ell)$ for allocating duration t_ℓ to the task ℓ, while she incurs a latency penalty c_ℓ per unit time for the delay in its processing. Therefore, the expected benefit for task ℓ is $w_\ell f_\ell(t_\ell) - c_\ell(t_1 + \ldots + t_\ell)$. The objective of the human operator is to maximize her average benefit and the associated optimization problem is:

$$\underset{\boldsymbol{t} \in \mathbb{R}^N_{\geq 0}}{\text{maximize}} \quad \frac{1}{N} \sum_{\ell=1}^{N} (w_\ell f_{d_\ell}(t_\ell) - (c_\ell + \cdots + c_N)t_\ell), \tag{4}$$

where $\boldsymbol{t} = \{t_1, \dots, t_N\}$ is the duration allocation vector.

4.2. Optimal solution

We start by establishing some properties of sigmoid functions. We study the optimization problem involving a sigmoid reward function and a linear latency penalty. In particular, given a sigmoid function f and a penalty rate $c \in \mathbb{R}_{>0}$, we wish to solve the following problem:

$$\underset{t \in \mathbb{R}_{\geq 0}}{\text{maximize}} \quad f(t) - ct. \tag{5}$$

The derivative of a sigmoid function is not a one-to-one mapping and hence, not invertible. We define the pseudo-inverse of the derivative of a sigmoid function f with inflection point t^{inf}, $f^\dagger : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ by

$$f^\dagger(y) = \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = y\}, & \text{if } y \in]0, f'(t^{\text{inf}})], \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Notice that the definition of the pseudo-inverse is consistent with Figure 2(a).

Lemma 1 (Sigmoid function and linear penalty)**.** *For the optimization problem* (5)*, the optimal solution t^* is*

$$t^* \in \operatorname{argmax}\{f(\beta) - c\beta \mid \beta \in \{0, f^\dagger(c)\}\}.$$

Proof. The global maximum lies at the point where first derivative is zero or at the boundary of the feasible set. The first derivative of the objective function is $f'(t) - c$. If $f'(t^{\text{inf}}) < c$, then the objective function is a decreasing function of time and the maximum is achieved at $t^* = 0$. Otherwise, a critical point is obtained by setting first derivative to zero. We note that $f'(t) = c$ has at most two roots. The second derivative condition yields that if there exist two roots, then only the larger of the two roots corresponds to a local maximum. Otherwise, the only root corresponds to a local maximum. The global maximum is determined by comparing the local maximum with the value of the objective function at the boundary $t = 0$. This completes the proof. □

Definition 1 (Critical penalty rate)**.** For a given sigmoid function f and penalty rate $c \in \mathbb{R}_{>0}$, let the solution of the problem (5) be $t^*_{f,c}$. The critical penalty rate ς_f is defined by

$$\varsigma_f = \sup\{c \in \mathbb{R}_{>0} \mid t^*_{f,c} \in \mathbb{R}_{>0}\}. \tag{7}$$

Note that the critical penalty rate is the slope of the tangent to the sigmoid function f from the origin. □

The optimal solution to problem (5) for different values of penalty rate c is shown in Figure 4. One may notice the optimal solution jumps down to zero at the critical penalty rate. This jump in the optimal allocation gives rise to combinatorial effects in the problems involving multiple sigmoid functions.

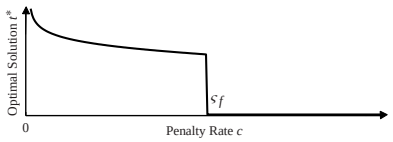


Figure 4: Optimal solution to the problem (5) as a function of linear penalty rate c. The optimal solution $t^* \to +\infty$ as the penalty rate $c \to 0^+$.

We can now analyze optimization problem (4).

Theorem 2 (Static queue with latency penalty)**.** *For the optimization problem* (4)*, the optimal allocation to task $\ell \in \{1, \dots, N\}$ is*

$$t^*_\ell \in \operatorname{argmax}\{w_\ell f_{d_\ell}(\beta) - (c_\ell + \cdots + c_N)\beta \mid \beta \in \{0, f^\dagger_{d_\ell}((c_\ell + \ldots + c_N)/w_\ell)\}\}.$$

Proof. The proof is similar to the proof of Lemma 1. □

Remark 2 (Comparison with a concave utility)**.** The optimal duration allocation for the static queue with latency penalty decreases to a critical value with increasing penalty rate, then jumps down to zero. In contrast, if the performance function is concave instead of sigmoid, then the optimal duration allocation decreases continuously to zero with increasing penalty rate. □

Example 1 (Static queue and homogeneous tasks)**.** The human operator has to serve $N = 10$ tasks and receives an expected reward $f(t) = 1/(1 + \exp(5 - t))$ for an allocation of duration t secs to a task, while she incurs a penalty $c = 0.02$ per sec for each pending task. The optimal policy according to Theorem 2 is shown in Figure 5(a). The optimal policy drops some tasks initially, then processes the remaining tasks. The duration allocation increases with decreasing number of pending tasks. □

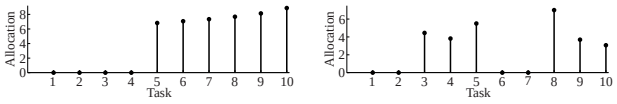
Example 2 (Static queue and heterogeneous tasks)**.** The human operator has to serve $N = 10$ heterogeneous tasks and receives an expected reward $f_{d_\ell}(t) = 1/(1 + \exp(-a_\ell t + b_\ell))$ for an allocation of duration t secs to task ℓ, where d_ℓ is characterized by the pair (a_ℓ, b_ℓ). The following are the parameters and the weights associated with each task:

$$\begin{aligned}(a_1, \ldots, a_N) &= (1, 2, 1, 3, 2, 4, 1, 5, 3, 6),\\ (b_1, \ldots, b_N) &= (5, 10, 3, 9, 8, 16, 6, 30, 6, 12), \text{ and}\\ (w_1, \ldots, w_N) &= (2, 5, 7, 4, 9, 3, 5, 10, 13, 6).\end{aligned}$$

Let the vector of penalty rates be

$$\boldsymbol{c} = (0.09, 0.21, 0.21, 0.06, 0.03, 0.15, 0.3, 0.09, 0.18, 0.06)$$

per second. The optimal allocations are shown in Figure 5(b). The importance and difficulty level of a task are encoded in the associated weight and the inflection point of the associated sigmoid function, respectively. The optimal allocations depend on the difficulty level, the penalty rate, and the importance of the tasks. For instance, task 6 is a relatively simple but less important task and is dropped. On the contrary, task 8 is a relatively difficult but very important task and is processed. □



(a) Homogeneous tasks (b) Heterogeneous tasks

Figure 5: Static queue with latency penalty. For homogeneous tasks, the optimal policy drops some tasks initially and then processes the remaining tasks. The duration allocation increases with decreasing queue length. For heterogeneous tasks, the optimal allocations depends of the difficulty level, the penalty rate and the importance of the tasks.

5. Dynamic queue with latency penalty: problem description and properties of optimal solution

In the previous section, we developed policies for static queue with latency penalty. We now consider dynamic queue with latency penalty, that is, the scenario where the tasks arrive according to a stochastic process and wait in a queue to get processed. We assume the tasks lose value while waiting in the queue. The operator's objective is to maximize her infinite horizon reward. In the following, we pose the problem as an MDP and show that the infinite horizon average value formulation of the MDP is equivalent to a deterministic dynamic optimization problem.

5.1. Problem description

Assume that the human operator has to serve a queue of decision making tasks arriving according to Poisson process with rate $\lambda \in \mathbb{R}_{>0}$. We assume that each task is sampled from a probability distribution function $p : \mathcal{D} \to \mathbb{R}_{\geq 0}$, where $\mathcal{D} \subseteq \mathbb{R}^q$ is the set of difficulty levels of the tasks. Each task is assigned a weight based on its importance. Two tasks that are equally difficult may have different weights. To capture this feature, we assume that the weight associated with a task with difficulty level d is a random variable $w_d \in \mathbb{R}_{>0}$ with probability distribution function $p_d^w : [w_d^{\min}, w_d^{\max}] \to \mathbb{R}_{\geq 0}$, where $w_d^{\min}, w_d^{\max} \in \mathbb{R}_{>0}$ are given constants. Similarly, let the latency penalty associated with a task with difficulty level d be a random variable $c_d \in \mathbb{R}_{>0}$ with probability distribution function $p_d^c : [c_d^{\min}, c_d^{\max}] \to \mathbb{R}_{\geq 0}$, where $c_d^{\min}, c_d^{\max} \in \mathbb{R}_{>0}$ are given constants. Let the realized difficulty level, importance, and latency penalty rate for task ℓ be d_ℓ, w_{d_ℓ}, and c_{d_ℓ}, respectively. Thus, the operator receives an expected reward $w_{d_\ell} f_{d_\ell}(t_\ell)$ for a duration allocation t_ℓ to task ℓ, while she incurs a latency penalty c_{d_ℓ} per unit time for the delay in its processing. Note that while designing a decision making queue, the true realizations of the random variables are not known and only expected values are at designer's disposal. Therefore, we construct the value function with the expected values over realizations of the queue. We define the expected reward function $\bar{f} : \mathbb{R}_{\geq 0} \to]0, 1[$ and the expected penalty rate $\bar{c} \in \mathbb{R}_{>0}$ by

$$\bar{f}(t) = \frac{1}{\bar{w}} \mathbb{E}_p[\mathbb{E}_{p_d^w}[w_d] f_d(t)] \text{ and } \bar{c} = \mathbb{E}_p[\mathbb{E}_{p_d^c}[c_d]],$$

respectively, where $\bar{w} = \mathbb{E}_p[\mathbb{E}_{p_d^w}[w_d]]$ and $\mathbb{E}_*[\cdot]$ represents the expected value with respect to the measure $*$. Note that these expressions assume that w_d, d, and c_d are statistically independent.

We denote the queue length at the beginning of processing task $\ell \in \mathbb{N}$ by $n_\ell \in \mathbb{Z}_{\geq 0}$. The objective of the operator is to maximize her infinite horizon expected reward. For any allocation t_ℓ to task ℓ, the queue length evolves according to a Poisson process and hence, it is Markovian. We now formulate the optimization problem as an MDP. We construct such an MDP, namely Γ, with the action space as the set of durations that can be allocated and the state space as the difficulty level of the tasks in the queue. We define the reward $r_\ell : \mathcal{D} \times \mathbb{R}^{n'_\ell} \times \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ obtained by allocating duration t to the task ℓ by

$$r_\ell(d_\ell, \boldsymbol{c}, w_{d_\ell}, t) = w_{d_\ell} f_{d_\ell}(t) - \frac{1}{2}\Big(\sum_{i=\ell}^{\ell+n_\ell-1} c_{d_i} + \sum_{j=\ell}^{\ell+n'_\ell-1} c_{d_j} \Big) t,$$

where $n'_\ell \in \mathbb{N}$ is the queue length just before the end of processing of the task $\ell \in \mathbb{N}$ and $\boldsymbol{c} \in \mathbb{R}^{n'_\ell}$ is the vector of penalty rates for the tasks in the queue. Note that the queue length while a task is processed may not be constant, therefore the latency penalty is computed as the average of the latency penalty for the tasks present at the start of processing the task and the latency penalty for the tasks present at the end of processing the task. Such averaging is consistent with expected number of arrivals being a linear function of time for Poisson process.

For a duration allocation t_ℓ to task $\ell \in \mathbb{N}$, the transition probability from queue length $n_\ell \in \mathbb{Z}_{\geq 0}$ to queue length $n_{\ell+1} \in \mathbb{Z}_{\geq 0}$ is

$$\mathbb{P}^{t_\ell}_{n_\ell n_{\ell+1}} = \begin{cases} 0, & \text{if } n_{\ell+1} \in \{0, \dots, n_\ell - 2\}, \\ e^{-\lambda t_\ell} \frac{(\lambda t_\ell)^{(n_{\ell+1}-n_\ell+1)}}{(n_{\ell+1}-n_\ell+1)!}, & \text{otherwise.} \end{cases}$$

The MDP with finite horizon length $N \in \mathbb{N}$ maximizes the value function $V_N : \mathbb{N} \times \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ defined by

$$V_N(n_1, \boldsymbol{t}) = \sum_{\ell=1}^{N} \mathbb{E}[r_\ell(d_\ell, \boldsymbol{c}, w_{d_\ell}, t_\ell)|n_1], \tag{8}$$

where $\boldsymbol{t}$ is the vector of allocations to each task and n_1 is the initial queue length.

The infinite horizon average value function of the MDP, denoted by $V_{\text{avg}} : \mathbb{N} \times \mathbb{R}^N_{\geq 0} \to \mathbb{R}$, is defined by

$$V_{\text{avg}}(n_1, \boldsymbol{t}) = \lim_{N \to +\infty} \frac{1}{N} V_N(n_1, \boldsymbol{t}).$$

We study the MDP Γ under the following assumptions:

Assumption 1 (Non-empty queue)**.** Without loss of generality, we assume that the queue is never empty. If queue is empty at some stage, then the operator waits for the next task to arrive, and there is no penalty for such waiting time.

Assumption 2 (Sigmoid average performance)**.** We assume the average of the sigmoid functions $\bar{f}$ is a sigmoid function.

Remark 3 (Sigmoid average performance)**.** Assumption 2 is justified in several contexts. For empirically obtained sigmoid functions, $\bar{f}$ can be obtained by fitting a sigmoid function through averaged empirical data. In the context of decision making tasks, the performance of the operator on each task is modeled by a drift-diffusion process, and the average of a set of drift-diffusion processes is again a drift-diffusion process. Hence, the average performance is well modeled by a sigmoid function. □

5.2. Properties of optimal solution

We now study some properties of the MDP Γ and its solution that will be used later in the paper. Before we establish these properties, we introduce the following optimization problem, which we refer to as the certainty-equivalent [1] problem:

$$\begin{aligned} \text{maximize} \quad & \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \Big(\bar{w}\bar{f}(t_\ell) - \bar{c}\mathbb{E}[n_\ell|n_1]t_\ell - \frac{\bar{c}\lambda t_\ell^2}{2} \Big) \\ \text{subject to} \quad & \mathbb{E}[n_{\ell+1}|n_1] = \max\{0, \mathbb{E}[n_\ell|n_1] - 1 + \lambda t_\ell\} \\ & t_\ell \geq 0, \forall \ell \in \mathbb{N}. \end{aligned} \tag{9}$$

We also define $N_{\max} = \lfloor \bar{w}\varsigma_{\bar{f}}/\bar{c} \rfloor$. We will show that $N_{\max}$ is the maximum queue length at which the optimal policy allocates non-zero duration to the first task. We now state some properties of the MDP Γ:

Lemma 3 (Properties of MDP Γ)**.** *Under Assumption 1 and 2, the following statements hold for the MDP Γ and its infinite horizon average value function:*

(i). *the MDP Γ admits the same optimal policy as* (9)*;*

(ii). *the optimal policy allocates zero duration to the first task if $n_1 > N_{\max}$;*

(iii). *the optimal policy allocates a duration less than $\bar{f}^\dagger(\bar{c}/\bar{w})$ to each task.*

Proof. We start with the definition of V_{avg}:

$$\begin{aligned} V_{\text{avg}}(n_1, \boldsymbol{t}) &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \mathbb{E}[r_\ell(d_\ell, n_\ell, n'_\ell, t_\ell)|n_1] \\ &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \mathbb{E}\Big[w_{d_\ell} f_{d_\ell}(t_\ell) - \frac{1}{2}\Big(\sum_{i=\ell}^{\ell+n_\ell-1} c_i + \sum_{j=\ell}^{\ell+n'_\ell-1} c_j \Big) t_\ell \Big| n_1 \Big] \\ &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \bar{w}\bar{f}(t_\ell) - \frac{1}{2}\bar{c}\mathbb{E}[n_\ell + n'_\ell|n_1]t_\ell \qquad (10) \\ &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \bar{w}\bar{f}(t_\ell) - \frac{1}{2}\bar{c}(2\mathbb{E}[(n_\ell|n_1] + \lambda t_\ell)t_\ell \\ &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \bar{w}\bar{f}(t_\ell) - \bar{c}\mathbb{E}[n_\ell|n_1]t_\ell - \frac{1}{2}\bar{c}\lambda t_\ell^2, \end{aligned}$$

where equation (10) follows from the Wald's identity [9] and the expected evolution of the queue length is determined by the Poisson arrival and the deterministic service processes, that is,

$$\mathbb{E}[n_{\ell+1}|n_1] = \max\{0, \mathbb{E}[n_\ell|n_1] - 1 + \lambda t_\ell\}, \ \forall \ell \in \{1, \dots, N\}.$$

Therefore, the infinite horizon average value formulation of the MDP and the certainty-equivalent problem are identical. This establishes the first statement.

To prove the second statement, we note that under Assumption 1, $\mathbb{E}[n_\ell|n_1] = n_1 - \ell + 1 + \lambda \sum_{j=1}^{\ell-1} t_j$ and thus, the value function is:

$$V_N(n_1, \boldsymbol{t}) = \sum_{\ell=1}^{N} \Big(\bar{w}\bar{f}(t_\ell) - \bar{c}(n_1 - \ell + 1)t_\ell - \bar{c}\lambda t_\ell \sum_{j=1}^{\ell-1} t_j - \frac{\bar{c}\lambda t_\ell^2}{2} \Big).$$

We write $V_N = V_{\text{one}} + V_{\text{rem}}$, where $V_{\text{one}} : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ and $V_{\text{rem}} : \mathbb{N} \times \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ are defined by

$$\begin{aligned} V_{\text{one}}(n_1, t_1) &= \bar{w}\bar{f}(t_1) - \bar{c}n_1 t_1, \\ V_{\text{rem}}(n_1, \boldsymbol{t}) &= \sum_{\ell=2}^{N} \Big(\bar{w}\bar{f}(t_\ell) - \bar{c}(n_1 - \ell + 1)t_\ell \\ &\qquad - \bar{c}\lambda t_\ell \sum_{j=1}^{\ell-1} t_j - \frac{\bar{c}\lambda t_\ell^2}{2} \Big) - \frac{c\lambda t_1^2}{2}. \end{aligned}$$

Note that V_{rem} is a decreasing function of t_1 and from Lemma 1 we know that, for $\bar{c}n_1/\bar{w} > \varsigma_{\bar{f}}$, V_{one} achieves its global maximum at $t_1 = 0$. Hence, V_N achieves its maximum at $t_1 = 0$ for $\bar{c}n_1/\bar{w} > \varsigma_{\bar{f}}$, that is, the optimal policy drops the first task if $n_1 > \bar{w}\varsigma_{\bar{f}}/\bar{c}$. Since, n_1 is a non-negative integer, $n_1 > \bar{w}\varsigma_{\bar{f}}/\bar{c}$ is equivalent to $n_1 > N_{\max}$.

To establish the last statement, we note that the function V_{one} is a decreasing function of t_1, for all $t_1 > \bar{f}^\dagger(\bar{c}/\bar{w})$, and V_{rem} is a

decreasing function of t_1, for all $t_1 > 0$. Hence the maximum allocation to any task is $\bar{f}^{\dagger}(\bar{c}/\bar{w})$. □

One of the key implications of Lemma 3 is that the solution of the MDP Γ is identical to the solution of a deterministic dynamic optimization problem. Although this reduces the computational complexity significantly, the computational cost to determine the optimal policy still grows exponentially with the size of the state space and the action space. We now exploit the results in Lemma 3 to reduce the dimensions of the state space and action space of the MDP Γ. We construct a reduced MDP, namely Γ_{red}, by restricting the action space to the possible allocations by the optimal policy and by aggregating all the queue lengths at which the optimal policy allocates zero duration to the current task into one state $N_{\max}+1$. Thus, picking the new action space as $[0, \bar{f}^{\dagger}(\bar{c}/\bar{w})]$, and the new state space as $\{0,\ldots,N_{\max}+1\}$. The new transition probabilities for allocating duration t_ℓ to task ℓ are defined by:

$$\mathbb{P}^{t_\ell}_{n_\ell n_{\ell+1}} = \begin{cases} 0, & \text{if } n_{\ell+1} \in \{0,\ldots,n_\ell-2\}, \\ e^{-\lambda t_\ell}\frac{(\lambda t_\ell)^{(n_{\ell+1}-n_\ell+1)}}{(n_{\ell+1}-n_\ell+1)!}, & \text{if } n_{\ell+1} \in \{n_\ell-1,\ldots,N_{\max}\}, \\ 1-\sum_{j=0}^{N_{\max}} \mathbb{P}^{t_\ell}_{n_\ell j}, & \text{if } n_{\ell+1} = N_{\max}+1. \end{cases}$$

The reward function $\bar{r}_\ell : \mathbb{N}\times\mathbb{R}_{\geq 0} \to \mathbb{R}$ for allocation of duration t_ℓ to task ℓ is defined by $\bar{r}_\ell(n_\ell, t_\ell) = \bar{w}\bar{f}(t_\ell) - \bar{c}n_\ell t_\ell - \bar{c}\lambda t_\ell^2/2$. We can now state the following equivalence.

Corollary 4 (Reducing the action space and the state space)**.** *The Markov decision processes* Γ *and* Γ_{red} *yield the same optimal policy.*

Proof. It can be verified that the value function for the two MDPs is the same. The reduction of the action space and the state space follows from Lemma 3. □

6. Dynamic queue with latency penalty: receding horizon algorithm

As discussed in the previous section, the computation of the optimal policy for infinite horizon average cost MDP problem (9) is expensive and grows exponentially with the dimension of the state space and action space. We rely on the receding horizon framework discussed in Section 2 to develop an approximation algorithm to determine the solution of the MDP Γ in finite time. As discussed in Algorithm 1, the receding horizon framework solves a finite horizon optimization problem at each iteration. We now study such finite horizon optimization problem for the certainty-equivalent problem (9).

6.1. Finite horizon optimization

We now study the finite horizon optimization problem with horizon length N that the receding horizon policy solves at each iteration. It follows from Lemma 3 that the MDP formulation is identical to the certainty-equivalent problem. Therefore, we focus on the solution of the finite horizon certainty-equivalent problem. Under Assumption 2, we treat $\bar{f}$ as a sigmoid function. For the ease of notation, we denote $\bar{f}$ and $\bar{c}/\bar{w}$ by f and c, respectively. We now introduce the following finite horizon optimization problem that needs to be solved at each stage in the receding horizon framework:

$$\begin{aligned} \underset{\boldsymbol{t}\geq 0}{\text{maximize}} \quad & \frac{1}{N}\sum_{\ell=1}^{N}\Big(f(t_\ell) - c\mathbb{E}[n_\ell|n_1]t_\ell - \frac{c\lambda t_\ell^2}{2}\Big) \\ \text{subject to} \quad & \mathbb{E}[n_{\ell+1}|n_1] = \max\{0, \mathbb{E}[n_\ell|n_1] - 1 + \lambda t_\ell\}, \end{aligned} \tag{11}$$

where $\boldsymbol{t} = \{t_1,\ldots,t_N\}$ is the duration allocation vector.

Under Assumption 1, the constraint in the optimization problem (11) yields:

$$\mathbb{E}[n_\ell|n_1] = n_1 - \ell + 1 + \lambda\sum_{j=1}^{\ell-1} t_j.$$

Substituting the expected queue length into the objective function in the optimization problem (11), one obtains the function $J : \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ defined by

$$J(\boldsymbol{t}) := \frac{1}{N}\sum_{\ell=1}^{N}\Big(f(t_\ell) - c(n_1-\ell+1)t_\ell - c\lambda t_\ell\sum_{j=1,j\neq\ell}^{N} t_j - \frac{c\lambda t_\ell^2}{2}\Big),$$

where c is the expected penalty rate, λ is the arrival rate, and n_1 is the initial queue length. Thus, the optimization problem (11) is equivalent to

$$\underset{\boldsymbol{t}\geq 0}{\text{maximize}} \quad J(\boldsymbol{t}). \tag{12}$$

In the remainder of Section 6.1, we propose a procedure to determine the solution of the optimization problem (11). To develop this procedure, we study some properties of the optimal policy. Assume that the solution to the optimization problem (11) allocates a strictly positive time only to the tasks in the set $\mathcal{T}_{\text{proc}} \subseteq \{1,\ldots,N\}$, which we call the *set of processed tasks*. (Accordingly, the policy allocates zero time to the tasks in $\{1,\ldots,N\}\setminus\mathcal{T}_{\text{proc}}$). Without loss of generality, assume

$$\mathcal{T}_{\text{proc}} := \{\eta_1,\ldots,\eta_m\},$$

where $\eta_1 < \cdots < \eta_m$ and $m \leq N$. A duration allocation vector $\boldsymbol{t}$ is said to be consistent with $\mathcal{T}_{\text{proc}}$ if only the tasks in $\mathcal{T}_{\text{proc}}$ are allocated non-zero duration.

Lemma 5 (Properties of maximum points)**.** *For the optimization problem* (12)*, and a set of processed tasks* $\mathcal{T}_{\text{proc}}$*, the following statements hold:*

(i). *a global maximum point* $\boldsymbol{t}^*$ *satisfy* $t^*_{\eta_1} \geq t^*_{\eta_2} \geq \ldots \geq t^*_{\eta_m}$*;*

(ii). *a local maximum point* $\boldsymbol{t}^{\dagger}$ *consistent with* $\mathcal{T}_{\text{proc}}$ *satisfies*

$$f'(t^{\dagger}_{\eta_k}) = c(n_1 - \eta_k + 1) + c\lambda\sum_{i=1}^{m} t^{\dagger}_{\eta_i}, \text{ for all } k \in \{1,\ldots,m\}; \tag{13}$$

(iii). *the system of equations* (13) *can be reduced to*

$$f'(t^{\dagger}_{\eta_1}) = \mathcal{P}(t^{\dagger}_{\eta_1}), \text{ and } t^{\dagger}_{\eta_k} = f^{\dagger}(f'(t^{\dagger}_{\eta_1}) - c(\eta_k - \eta_1)),$$

for each $k \in \{2, \dots, m\}$, *where* $\mathcal{P} : \mathbb{R}_{>0} \to \mathbb{R} \cup \{+\infty\}$ *is defined by*

$$\mathcal{P}(t) = \begin{cases} p(t), & \text{if } f'(t) \geq c(\eta_m - \eta_1), \\ +\infty, & \text{otherwise,} \end{cases}$$

where $p(t) = c(n_1 - \eta_1 + 1 + \lambda t + \lambda \sum_{k=2}^{m} f^{\dagger}(f'(t) - c(\eta_k - \eta_1)))$;

(iv). *a local maximum point* $\boldsymbol{t}^{\dagger}$ *consistent with* $\mathcal{T}_{\text{proc}}$ *satisfies*

$$f''(t_{\eta_k}) \leq c\lambda, \quad \text{for all} \quad k \in \{1, \dots, m\}.$$

Proof. We start by proving the first statement. Assume $t^*_{\eta_j} < t^*_{\eta_k}$ and define the allocation vector $\bar{\boldsymbol{t}}$ consistent with $\mathcal{T}_{\text{proc}}$ by

$$\bar{t}_{\eta_i} = \begin{cases} t^*_{\eta_i}, & \text{if } i \in \{1, \dots, m\} \setminus \{j, k\}, \\ t^*_{\eta_j}, & \text{if } i = k, \\ t^*_{\eta_k}, & \text{if } i = j. \end{cases}$$

It is easy to see that

$$J(\boldsymbol{t}^*) - J(\bar{\boldsymbol{t}}) = (\eta_j - \eta_k)(t^*_{\eta_j} - t^*_{\eta_k}) < 0.$$

This inequality contradicts the assumption that $\boldsymbol{t}^*$ is a global maximum of J.

To prove the second statement, note that a local maximum is achieved at the boundary of the feasible region or at the set where the Jacobian of J is zero. At the boundary of the feasible region $\mathbb{R}^N_{\geq 0}$, some of the allocations are zero. Given the m non-zero allocations, the Jacobian of the function J projected on the space spanned by the non-zero allocations must be zero. The expressions in the theorem are obtained by setting the Jacobian to zero.

To prove the third statement, we subtract the expression in equation (13) for $k = j$ from the expression for $k = 1$ to get

$$f'(t_{\eta_j}) = f'(t_{\eta_1}) - c(\eta_j - \eta_1). \tag{14}$$

There exists a solution of equation (14) if and only if $f'(t_{\eta_1}) \geq c(\eta_j - \eta_1)$. If $f'(t_{\eta_1}) < c(\eta_j - \eta_1) + f'(0)$, then there exists only one solution. Otherwise, there exist two solutions. It can be seen that if there exist two solutions $t^{\pm}_j$, with $t^-_j < t^+_j$, then $t^-_j < t_{\eta_1} < t^+_j$. From the first statement, it follows that only possible allocation is t^+_j. Notice that $t^+_j = f^{\dagger}(f'(t_{\eta_1}) - c(\eta_j - \eta_1))$. This choice yields feasible time allocation to each task $\eta_j, j \in \{2, \dots, m\}$ parametrized by the time allocation to the task η_1. A typical allocation is shown in Figure 6(a). We further note that the effective penalty rate for the task η_1 is $c(n_1 - \eta_1 + 1) + c\lambda \sum_{j=1}^{m} t_{\eta_j}$. Using the expression of $t_{\eta_j}, j \in \{2, \dots, m\}$, parametrized by t_{η_1}, we obtain the expression for $\mathcal{P}$.

To prove the last statement, we observe that the Hessian of the function J is

$$\frac{\partial^2 J}{\partial \boldsymbol{t}^2} = \text{diag}(f''(t_{\eta_1}), \dots, f''(t_{\eta_m})) - c\lambda \mathbf{1}_m \mathbf{1}_m^T,$$

where diag$(\cdot)$ represents a diagonal matrix with the argument as diagonal entries. For a local maximum to exist at non-zero duration allocations $\{t_{\eta_1}, \dots, t_{\eta_m}\}$, the Hessian must be negative semidefinite. A necessary condition for Hessian to be negative semidefinite is that diagonal entries are non-positive. □

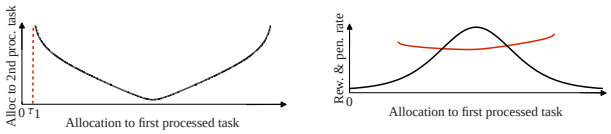
We refer to the function $\mathcal{P}$ as the *effective penalty rate* for the first processed task. A typical graph of $\mathcal{P}$ is shown in Figure 6(b). Given $\mathcal{T}_{\text{proc}}$, a feasible allocation to the task η_1 is such that $f'(t_{\eta_1}) - c(\eta_j - \eta_1) > 0$, for each $j \in \{2, \dots, m\}$. For a given $\mathcal{T}_{\text{proc}}$, we define the minimum feasible duration allocated to task η_1 (see Figure 6(a)) by

$$\tau_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = c(\eta_m - \eta_1)\}, & \text{if } f'(t^{\text{inf}}) \geq c(\eta_m - \eta_1), \\ 0, & \text{otherwise.} \end{cases}$$

Let $f''_{\max}$ be the maximum value of f''. We now define the points at which the function $f'' - c\lambda$ changes its sign (see Figure 2(b)):

$$\delta_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \in [f''(0), f''_{\max}], \\ 0, & \text{otherwise,} \end{cases}$$

$$\delta_2 := \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \leq f''_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$



(a) (b)

Figure 6: (a) Feasible allocations to the second processed task parametrized by the allocation to the first processed task. (b) The penalty rate and the sigmoid derivative as a function of the allocation to the first task.

Theorem 6 (Finite horizon optimization)**.** *Given the optimization problem* (12)*, and a set of processed tasks* $\mathcal{T}_{\text{proc}}$*. The following statements are equivalent:*

(i). *there exists a local maximum point consistent with* $\mathcal{T}_{\text{proc}}$*;*

(ii). *one of the following conditions hold*

$$f'(\delta_2) \geq \mathcal{P}(\delta_2), \quad \text{or} \tag{15}$$

$$f'(\tau_1) \leq \mathcal{P}(\tau_1), \; f'(\delta_1) \geq \mathcal{P}(\delta_1), \; \text{and } \delta_1 \geq \tau_1. \tag{16}$$

Proof. A critical allocation to task η_1 is located at the intersection of the graph of the reward rate $f'(t_{\eta_1})$ and the effective penalty rate $\mathcal{P}(t_{\eta_1})$. From Lemma 5, a necessary condition for the existence of a local maximum at a critical point is $f''(t_{\eta_1}) \leq c\lambda$, which holds for $t_{\eta_1} \in]0, \delta_1] \cup [\delta_2, \infty[$. It can be seen that if condition (15) holds, then the function $f'(t_{\eta_1})$ and the effective penalty function $\mathcal{P}(t_{\eta_1})$ intersect in the region $[\delta_2, \infty[$. Similarly, condition (16) ensures the intersection of the graph of the reward function $f'(t_{\eta_1})$ with the effective penalty function $\mathcal{P}(t_{\eta_1})$ in the region $]0, \delta_1]$. □

We now provide a procedure to determine the solution to the optimization problem (12). Given a sequence of zero and non-zero allocations $\xi \in \{0, +\}^N$, we denote the corresponding critical allocation for maximum by $\boldsymbol{t}(\xi)$. The details of the procedure are shown in Algorithm 2. We refer to the policy obtained from receding horizon algorithm that solves the optimization problem (12) at each stage as the *certainty-equivalent policy.*

Algorithm 2 Finite horizon allocation algorithm

given n_1, N, c, λ
$k := 0$; $\mathcal{A} := \phi$;
for each string $\xi \in \{0, +\}^N$
  set $\mathcal{T}_{\text{proc}} := \{i \in \{1, \ldots, N\} \mid \xi_i = +\}$
  if condition (15) or (16)
  then determine critical allocations for maximum $t^{\dagger}_{\eta_1}$ via bisection algorithm
    determine allocations $t^{\dagger}_{\eta_j} = \overline{f}^{\dagger}(\overline{f}'(t_{\eta_1}) - c(\eta_j - \eta_1))$, $j \in \{2, \ldots, m\}$
    determine expected queue length $\mathbb{E}[n_\ell]$, $\ell \in \{1, \ldots, N\}$
      if $\mathbb{E}[n_\ell] > 0, \forall \ell \in \{1, \ldots, N\}$
      then $\mathcal{A} = \mathcal{A} \cup \{\boldsymbol{t}^{\dagger}(\xi)\}$
optimal allocation $\boldsymbol{t}^* = \operatorname{argmax}_{\boldsymbol{t} \in \mathcal{A}} J(\boldsymbol{t})$

Remark 4 (Computational complexity of Algorithm 2)**.** In the worst case, Algorithm 2 requires a comparison of the solution of $2^N - 1$ optimization problems. Although the number of worst-case comparisons grows exponentially with the chosen horizon length N, it remains reasonable for fairly large horizon lengths ($N \leq 10$). □

Remark 5 (Comparison with a concave utility)**.** With the increasing penalty rate as well as the increasing arrival rate, the time duration allocation decreases to a critical value and then jumps down to zero, for the dynamic queue with latency penalty. In contrast, if the performance function is concave instead of sigmoid, then the duration allocation decreases continuously to zero with increasing penalty rate as well as increasing arrival rate. □

6.2. Performance of receding horizon algorithm

We now derive performance bounds on the certainty-equivalent policy. First, we determine a global upper bound on the performance of any policy for the MDP Γ. Then, we develop a lower bound on the performance of the unit horizon certainty-equivalent policy, that is, the policy obtained from the receding horizon algorithm that solves optimization problem (11) with horizon length $N = 1$ at each iteration. The performance of the unit horizon certainty-equivalent policy provides a lower bound to the performance of any certainty-equivalent policy that solves a finite horizon problem with horizon length $N > 1$ at each stage. Let $\boldsymbol{t}^{\text{rec}}$ be the sequence of duration allocations under a certainty-equivalent policy. Without loss of generality, we assume that the initial queue length is unity. If the initial queue length is non-unity, then we drop tasks till queue length is unity. Note that this does not affect the infinite horizon average value function. We also assume that the latency penalty is small enough to ensure an optimal non-zero duration allocation if only one task is present in the queue, that is, $\overline{c} \leq \overline{w}\varsigma_{\overline{f}}$. We now derive a lower bound on the performance of the unit horizon certainty-equivalent policy, which is also a lower bound on the performance of any certainty-equivalent policy.

Theorem 7 (Bounds on performance)**.** *For the Markov Decision Process* Γ*, and any certainty-equivalent policy the following statements hold, provided* $\overline{c} \leq \overline{w}\varsigma_{\overline{f}}$*:*

(i). *the average value function satisfy the following upper bound*

$$V_{\text{avg}}(n_1, \boldsymbol{t}) \leq \overline{w}\overline{f}(\overline{f}^{\dagger}(\overline{c}/\overline{w})) - \overline{c}\overline{f}^{\dagger}(\overline{c}/\overline{w}),$$

for each $n_1 \in \mathbb{N}$ *and any non-negative sequence* $\boldsymbol{t}$*;*

(ii). *the average value function satisfy the following lower bound for any certainty-equivalent policy:*

$$V_{\text{avg}}(n_1, \boldsymbol{t}^{\text{rec}}) \geq \begin{cases} \overline{w}\overline{f}(\tau_{\max}) - \overline{c}\tau_{\max} - \frac{\overline{c}\lambda\tau_{\max}^2}{2}, & \text{if } 0 < \lambda \leq \frac{1}{\tau_{\max}}, \\ \frac{1}{\lceil \lambda\tau_{\max} \rceil}(\overline{w}\overline{f}(\tau_{\min}) - \varsigma_{\overline{f}}\tau_{\max} - \frac{\overline{c}\lambda\tau_{\max}^2}{2}), & \text{otherwise,} \end{cases}$$

for each $n_1 \in \mathbb{N}$*, where* $\tau_{\max} = \overline{f}^{\dagger}(\overline{c}/\overline{w})$ *and* $\tau_{\min} = \overline{f}^{\dagger}(\varsigma_{\overline{f}})$*.*

Proof. We start by establishing the first statement. We recall from Lemma 3 that the value function V_{avg} is identical to the objective function of the certainty-equivalent problem (9), that is,

$$\begin{aligned} V_{\text{avg}}(n_1, \boldsymbol{t}) &= \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \overline{w}\overline{f}(t_\ell) - \overline{c}\mathbb{E}[n_\ell | n_1] t_\ell - \overline{c}\lambda t_\ell^2/2 \\ &\leq \lim_{N \to +\infty} \frac{1}{N} \sum_{\ell=1}^{N} \overline{w}\overline{f}(t_\ell) - \overline{c}t_\ell \\ &\leq \overline{w}\overline{f}(\overline{f}^{\dagger}(\overline{c}/\overline{w})) - \overline{c}\overline{f}^{\dagger}(\overline{c}/\overline{w}), \end{aligned}$$

where the last inequality follows from Lemma 1.

In order to determine a lower bound, we construct following allocation policy:

$$t_\ell^{\text{low}} = \begin{cases} \tau_{\max}, & 0 < \lambda \leq 1/\tau_{\max}, \\ t_\ell^{\text{stat}}, & \text{otherwise,} \end{cases}$$

for each $\ell \in \mathbb{N}$, where $t_\ell^{\text{stat}} \in \operatorname{argmax}\{\overline{w}\overline{f}(\beta) - \overline{c}\overline{n}_\ell\beta \mid \beta \in \{0, \overline{f}^{\dagger}(\overline{n}_\ell\overline{c}/\overline{w})\}\}$ and $\overline{n}_\ell = \mathbb{E}[n_\ell | n_1]$. We note that the unit horizon certainty-equivalent policy allocates duration $t_\ell^{\text{unit}} \in \operatorname{argmax}\{\overline{w}\overline{f}(t) - \overline{c}\overline{n}_\ell t - \overline{c}\lambda t^2/2 \mid t \in \mathbb{R}_{\geq 0}\}$ to task $\ell \in \mathbb{N}$. Therefore,

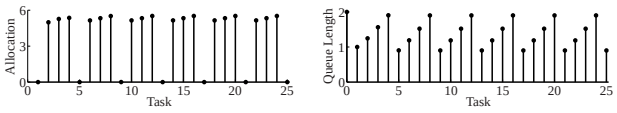
$$\begin{aligned} &\overline{w}\overline{f}(t_\ell^{\text{unit}}) - \overline{c}\overline{n}_\ell t_\ell^{\text{unit}} - \overline{c}\lambda {t_\ell^{\text{unit}}}^2/2 \geq \overline{w}\overline{f}(t_\ell^{\text{low}}) - \overline{c}\overline{n}_\ell t_\ell^{\text{low}} - \overline{c}\lambda {t_\ell^{\text{low}}}^2/2 \\ &\qquad \implies V_{\text{avg}}(n_1, \boldsymbol{t}^{\text{unit}}) \geq V_{\text{avg}}(n_1, \boldsymbol{t}^{\text{low}}). \end{aligned}$$

We first consider the case when $0 < \lambda \leq 1/\tau_{\max}$. The constructed policy allocates duration $\tau_{\max}$ to each task. For the certainty-equivalent problem, a new task arrives in time $1/\lambda \geq \tau_{\max}$, that is, after servicing the current task, the queue is either empty or has one task. Therefore, the expected reward for each task is $\overline{w}\overline{f}(\tau_{\max}) - \overline{c}\tau_{\max} - \overline{c}\lambda\tau_{\max}^2/2$.

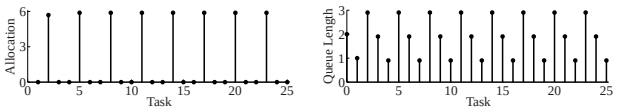
In the second case, we note that the maximum allocation to each task under the constructed policy is $\tau_{\max}$ and hence, the maximum number of expected arrivals while processing current task is $\lambda\tau_{\max}$. In the worst possible case, $\lceil \lambda\tau_{\max} \rceil - 1$ tasks would be dropped before next task is served. Further, the duration allocation to the task is in the interval $[\tau_{\min}, \tau_{\max}]$ and the penalty $\overline{c} \leq \overline{w}\varsigma_{\overline{f}}$. Thus, the lower bound follows. □

We now elucidate on the concepts discussed in this section with an example.

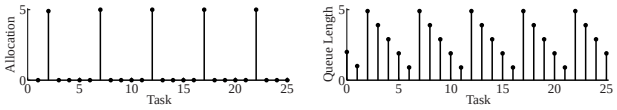
Example 3 (Certainty-equivalent policy)**.** Suppose that the human operator has to serve a queue of tasks with Poisson arrival at the rate λ per sec. The set of the tasks is the same as in Example 2 and each task is sampled uniformly from this set. For this set of data, the average performance function is $\bar{f}(t) = \bar{w}/(1 + e^{-at+b})$, where $\bar{w} = 6.4$, $a = 1.0853$, and $b = 4.3027$. The average penalty rate is $\bar{c} = 0.1380$ per second. The certainty-equivalent policy that solves problem (11) with horizon length $N = 10$ at each stage is shown in Figure 7. It can be seen that the certainty-equivalent policy drops more tasks at higher arrival rates and tries to maintain a single task in the queue. The performance of the certainty-equivalent policy along with the global upper bound on the performance of any policy and the lower bound on the performance of any certainty-equivalent policy is shown in Figure 8. As expected, for the low arrival rates the certainty-equivalent policy achieves a performance very close to the global upper bound. □



(a) Low arrival rate



(b) Moderate arrival rate



(c) High arrival rate

Figure 7: Certainty-equivalent policy. An optimization problem with horizon length $N = 10$ is solved at each stage. The arrival rates for the three scenarios are $\lambda = 0.25, 0.5$ and 1, respectively.

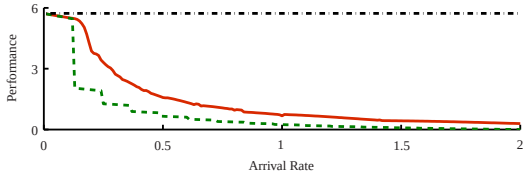


Figure 8: Bounds on performance. The solid red curve represents the average value function under certainty-equivalent policy, the dashed-dotted black line represents the upper bound on any policy and the dashed green curve represents the lower bound on any certainty-equivalent policy.

Discussion 8 (Optimal arrival rate)**.** The performance of the certainty-equivalent policy as a function of the arrival rate is shown in Figure 9. It can be seen that the expected benefit per unit task, that is, the value of the average value function under the certainty-equivalent policy, decreases slowly till a critical arrival rate and then starts decreasing quickly. This critical arrival rate corresponds to the situation where a new task is expected to arrive as soon as the operator finishes processing the current task. For the set of data considered, the benefit per unit time achieves its maximum at this critical arrival rate. In general, it is not true and this maximum may be achieved at a value higher than the critical arrival rate. Thus, the arrival rate maximizing benefit per unit time may result in poor average decision quality on each task. The objective of the designer is to achieve a good performance on each task and therefore, the arrival rate should be picked close to the critical arrival rate. It can be verified that the critical arrival rate is $\lambda_{\text{crit}} = 1/\bar{f}^{-1}(2\bar{c}/\bar{w})$. In general, there may be other performance goals for the operator, and accordingly, higher task arrival rate for the queue could be designed. □

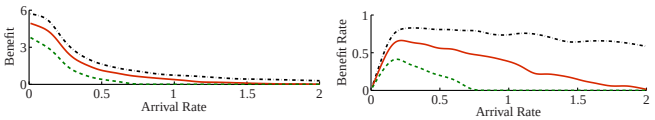


Figure 9: Expected benefit per unit task and per unit time over a finite horizon under certainty-equivalent policy. The dashed-dotted black, solid red and dashed green curves correspond to latency penalties 0.01, 0.025, and 0.05, respectively.

7. Dynamic queue with latency penalty: receding horizon algorithm with real time information

We studied the receding horizon policies for the certainty-equivalent problem which is identical to infinite horizon average cost formulation of the underlying MDP. While designing the decision making queue, the true realization of the tasks and the associated latency penalty and importance is not known. Therefore, the policy is designed for the expected evolution of the queue. In particular, the computation of the value function in equation (8) involved the expectation over realizations of the queue. In real time, the information about the nature of the current tasks in the queue is available and should be incorporated in the value function. We incorporate this information in the following way. We define new value function $V_N^{\text{rlzd}} : \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{N} \to \mathbb{R}$ by

$$V_N^{\text{rlzd}}(\boldsymbol{d}, \boldsymbol{w}, \boldsymbol{C}, \boldsymbol{t}) = \sum_{\ell=1}^{N} \mathbb{E}[r_\ell(d_\ell, \boldsymbol{c}, w_{d_\ell}, t_\ell)|\mathcal{F}_\ell],$$

where $\mathbb{R}_{\geq 0}^{\infty}$ represents sequences of positive real numbers, $\mathcal{F}_\ell$ represents the sigma algebra containing all the information available when task ℓ is processed, $\boldsymbol{d}, \boldsymbol{w}$, and $\boldsymbol{C}$ are the sequences of realized difficulty levels, weights, and latency penalties, respectively.

With the real time information, the infinite horizon average value function of the MDP $V_{\text{avg}}^{\text{rlzd}} : \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{\infty} \times \mathbb{R}_{\geq 0}^{L} \to \mathbb{R}$ is defined by

$$V_{\text{avg}}^{\text{rlzd}}(\boldsymbol{d}, \boldsymbol{w}, \boldsymbol{C}, \boldsymbol{t}) = \lim_{N \to +\infty} \frac{1}{N} V_N^{\text{rlzd}}(\boldsymbol{d}, \boldsymbol{w}, \boldsymbol{C}, \boldsymbol{t}).$$

In the spirit of Section 6, we develop receding horizon algorithms to maximize $V_{\text{avg}}^{\text{rlzd}}$. We solve the associated finite horizon problem using dynamic programming with discretized action and state space.

Remark 6 (Finite horizon problem)**.** It can be verified that the finite horizon problem associated with the maximization of $V_{\text{avg}}^{\text{rlzd}}$ is similar to the optimization problem (11), but due to the non-identical nature of the tasks, the allocations to the processed tasks can not be parametrized as a function of the allocation to the first processed task (see Lemma 5). Thus, the search for the optimal allocation can not be reduced to a one dimensional search. This makes the extension of the techniques in Section 6 to the maximization of $V_{\text{avg}}^{\text{rlzd}}$ intractable. Therefore, we utilize dynamic programming with discretized action and state space to approximately solve the finite horizon problem.□

Before we present the receding horizon algorithm, we introduce few notations. An analogous argument to the one in Lemma 3 shows that under optimal policy the maximum allocation to a sigmoid function f with latency penalty c and weight w is upper bounded by $f^{\dagger}(c/w)$. We define the maximum allocation to any sigmoid function by $\delta_{\max} = \sup\{f_d^{\dagger}(c_d^{\min}/w_d^{\max}) \mid d \in \mathcal{D}\}$. Given horizon length N, current queue length $n_\ell \leq N$, the realization of the sigmoid functions $f_1, \ldots, f_{n_\ell}$, the associated latency penalties $c_1, \ldots, c_{n_\ell}$ and importance $w_1, \ldots, w_{n_\ell}$, we define the reward associated with task $j \in \{1, \ldots, N\}$ by

$$r_j = \begin{cases} r_j^{\text{rlzd}}, & \text{if } 1 \leq j \leq n_\ell, \\ r_j^{\text{exp}}, & \text{if } n_\ell + 1 \leq j \leq N, \end{cases} \tag{17}$$

where $r_j^{\text{rlzd}} = w_j f_j(t_j) - (\sum_{i=j}^{n_\ell} c_i + (\mathbb{E}[n_j] - n_\ell - j + 1)\bar{c})t_j - \bar{c}\lambda t_j^2/2$, and $r_j^{\text{exp}} = \bar{w} f(t_j) - \bar{c}(n_\ell - j + 1)t_j - \bar{c}\lambda t_j^2/2$. We now formally introduce this dynamic programming based algorithm in Algorithm 3, and refer to it as *adaptive allocation algorithm*. This algorithm incorporates the precise information of the tasks currently waiting in the queue while processing each task and thus adapts the allocation policy as new information becomes available. We will now provide numerical evidence to show that adaptive allocation policy improves the performance over the policies discussed in Section 6.

Algorithm 3 Adaptive Allocation Algorithm

1: **Given:** $f_d, d \in \mathcal{D}$, horizon length N, arrival rate λ, set $\ell = 1$
2: For task ℓ determine queue length n_ℓ, sigmoid functions and penalty rates f_i, c_i for each task $i \in \{1, \ldots, n_\ell\}$
3: **if** $n_\ell < N$
4: set stage rewards r_j using equation (17), $\forall\, j \in \{1, \ldots, N\}$,
5: **else** set stage rewards, for each $j \in \{1, \ldots, N\}$,
$r_j = w_j f_j(t_j) - (\sum_{i=j}^{N} c_i + (\mathbb{E}[n_j] - n_\ell - j + 1)\bar{c})t_j - \bar{c}\lambda t_j^2/2$.
6: solve the finite horizon DP with appropriately discretized allocations $t_j \in [0, \delta_{\max}]$, for each $j \in \{1, \ldots, N\}$
7: allocate duration t_1 to the task ℓ
8: set $\ell = \ell + 1$ and go to step 2:

Example 4 (Adaptive allocation policy)**.** For the data in Example 3, we now study the adaptive allocation policy. Adaptive allocation policies with horizon length 1 and 10 for a sample evolution of the queue at an arrival rate $\lambda = 0.5$ per second are shown in Figure 10 and 11, respectively. The adaptive policy tends to drop the tasks that are difficult and unimportant. The difficulty of the tasks is characterized by the inflection point of the associated sigmoid functions. Due to the heterogeneous nature of the tasks, the queue length under the adaptive policy is larger than the queue length under certainty-equivalent policy. The queue length under the adaptive allocation policy with horizon length 1 is higher than the adaptive allocation policy with horizon length 10. A comparison of the certainty-equivalent policy and the adaptive allocation policies is shown in Figure 12. We obtained these performance curves through Monte-Carlo simulations. It can be seen that the adaptive allocation policy improves the performance significantly over the certainty-equivalent policy. Interestingly, the performance of the adaptive allocation policy with horizon length $N = 1$ is also better than the certainty-equivalent policy. Thus, incorporating the available information significantly improves the performance. □

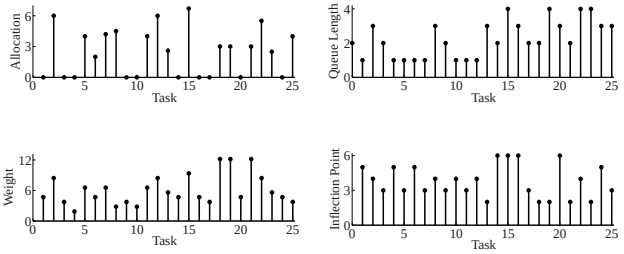


Figure 10: Adaptive policy for a *sample evolution* of the dynamic queue with latency penalty. An optimization problem with horizon length $N = 10$ is solved at each stage.

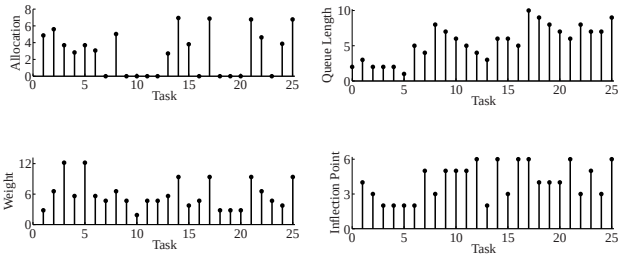


Figure 11: Adaptive policy for a *sample evolution* of the dynamic queue with latency penalty. An optimization problem with horizon length $N = 1$ is solved at each stage.

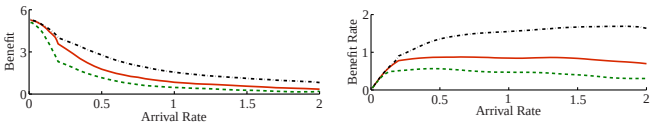


Figure 12: Empirical expected benefit per unit task and per unit time. The dashed-dotted black curve represents the adaptive allocation policy with horizon length 10, the solid red curve represents the adaptive allocation policy with horizon length 1, and the dashed green curve represents the certainty-equivalent policy with horizon length 10, respectively.

8. Conclusions

We presented optimal servicing policies for the queues where the performance function of the server is a sigmoid function. First, we considered a queue with no arrival and a latency penalty. It was observed that the optimal policy may drop some tasks. Further, for identical tasks, the duration allocation to the task increases with the decreasing queue length. Second, a dynamic queue with latency penalty was considered. We first studied the scenario where no real time information about the evolution of the queue was available. This models the situation of the designer who has no information about the true realization of queue at her disposal. A receding horizon algorithm was established for the certainty-equivalent problem and guidelines for choosing the arrival rate were suggested. We then studied the scenario where real time information about the realization of the queue was available. An adaptive allocation algorithm that incorporated all the available information about the current tasks into the allocation policy was developed. A comparison of the certainty-equivalent policy and the adaptive allocation policy was presented.

The decision support system designed in this paper assumes that the arrival rate of the tasks as well as the parameters in the performance function are known. An interesting open problem is to come up with policies which perform an online estimation of the arrival rate and the parameters of the performance function and simultaneously determine the optimal allocation policy. Another interesting problem is to incorporate more human factors into the optimal policy, for example, situational awareness, fatigue, etc. The policies designed in this paper rely on first-come first-serve discipline to process tasks. It would be of interest to study problems with other processing disciples, for example, preemptive queues.

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[2] L. F. Bertuccelli, N. W. M. Beckers, and M. L. Cummings. Developing operator models for UAV search scheduling. In *AIAA Conf. on Guidance, Navigation and Control*, Toronto, Canada, August 2010.

[3] L. F. Bertuccelli, N. Pellegrino, and M. L. Cummings. Choice modeling of relook tasks for UAV search missions. In *American Control Conference*, pages 2410–2415, Baltimore, MD, USA, USA, June 2010.

[4] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen. The physics of optimal decision making: A formal analysis of performance in two-alternative forced choice tasks. *Psychological Review*, 113(4):700–765, 2006.

[5] W. M. Bulkeley. Chicago's camera network is everywhere. The Wall Street Journal, Nov 17 2009.

[6] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2004.

[7] C. Drew. Military taps social networking skills. The New York Times, June 7, 2010.

[8] J. M. George and J. M. Harrison. Dynamic control of a queue with adjustable service rate. *Operations Research*, 49(5):720–731, 2001.

[9] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

[10] E. Guizzo. Obama commanding robot revolution announces major robotics initiative. IEEE Spectrum, June 2011.

[11] O. Hernández-Lerma and S. I. Marcus. Adaptive control of service in queueing systems. *Systems & Control Letters*, 3(5):283–289, 1983.

[12] S. K. Hong and C. G. Drury. Sensitivity and validity of visual search models for multiple targets. *Theoretical Issues in Ergonomics Science*, 3(1):85–110, 2002.

[13] S. Stidham Jr. and R. R. Weber. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations Research*, 37(4):611–625, 1989.

[14] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[15] R. W. Pew. The speed-accuracy operating characteristic. *Acta Psychologica*, 30:16–26, 1969.

[16] N. D. Powel and K. A. Morgansen. Multiserver queueing for supervisory control of autonomous vehicles. In *American Control Conference*, Montréal, Canada, June 2012. To appear.

[17] M. H. Rothkopf. Bidding in simultaneous auctions with a constraint on exposure. *Operations Research*, 25(4):620–629, 1977.

[18] K. Savla and E. Frazzoli. Maximally stabilizing task release control policy for a dynamical queue. *IEEE Transactions on Automatic Control*, 55(11):2655–2660, 2010.

[19] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proceedings of the IEEE*, 100(3), 2012. To appear.

[20] K. Savla, T. Temple, and E. Frazzoli. Human-in-the-loop vehicle routing policies for dynamic environments. In *IEEE Conf. on Decision and Control*, pages 1145–1150, Cancún, México, December 2008.

[21] D. K. Schmidt. A queuing analysis of the air traffic controller's work load. *IEEE Transactions on Systems, Man & Cybernetics*, 8(6):492–498, 1978.

[22] L. I. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley, 1999.

[23] T. Shanker and M. Richtel. In new military, data overload can be deadly. The New York Times, Jan 16, 2011.

[24] V. Srivastava and F. Bullo. Hybrid combinatorial optimization: Sample problems and algorithms. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7212–7217, Orlando, FL, USA, December 2011.

[25] V. Srivastava, R. Carli, F. Bullo, and C. Langbort. Task release control for decision making queues. In *American Control Conference*, pages 1855–1860, San Francisco, CA, USA, June 2011.

[26] D. Vakratsas, F. M. Feinberg, F. M. Bass, and G. Kalyanaram. The shape of advertising response functions revisited: A model of dynamic probabilistic thresholds. *Marketing Science*, 23(1):109–119, 2004.

[27] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, third edition, 2000.